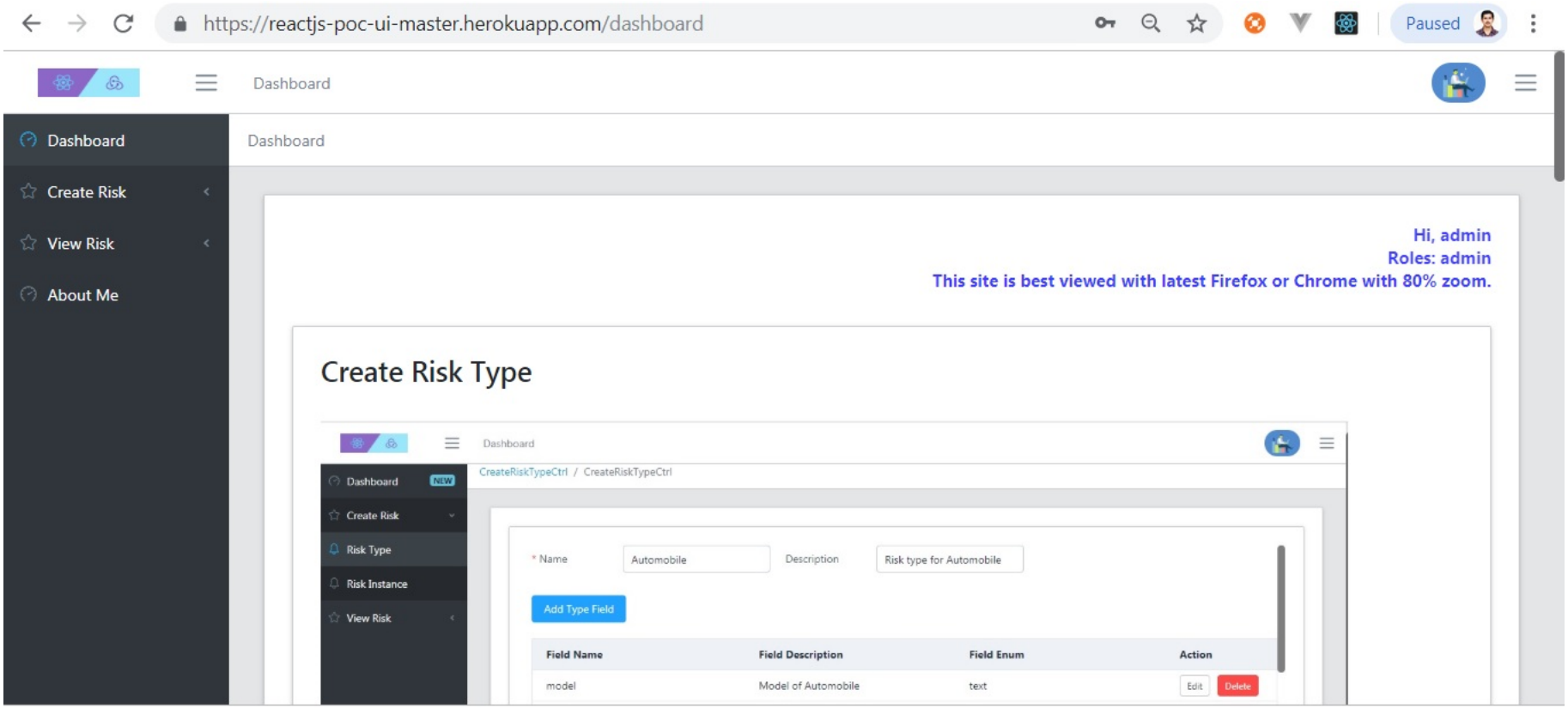
# Home Screen


https://reactjs-poc-ui-master.herokuapp.com/dashboard
Paused
Dashboard
Dashboard
Create Risk
View Risk
About Me
Dashboard
Hi, admin
Roles: admin
This site is best viewed with latest Firefox or Chrome with 80% zoom.
Create Risk Type
Dashboard
NEW
Create Risk
Risk Type
Risk Instance
View Risk
CreateRiskTypeCtrl / CreateRiskTypeCtrl
* Name
Automobile
Description
Risk type for Automobile
Add Type Field
Field Name
Field Description
Field Enum
Action
model
Model of Automobile
text
Edit
Delete

# ReactJS PoC UI

- https://reactjs-poc-ui-master.herokuapp.com/login
- Use above link to access UI.
- Username :- admin (lower case)
- Password :- poctest#1  (lower case)
- Best viewed with Firefox with 80 % zoom
- Above UI refers Web API at following location credentials are same.
- https://django-poc-server-maheshbodas.herokuapp.com/

# Salient features

- Developed using ReactJS, Redux, redux-thunk, ES6, Element React library, Node JS, Serve as a Static server.
- Used Redux as global data store for this application. Redux actions and reducers make is easy to shape and store data as needed by UI.
- Used CoreUI React template so as to reuse Sidebar navigation, Breadcrumbs and common Layout for all pages.

# Salient features (continued)

- CoreUI React template itself build using Create React App. CRA is zero configuration template to build, unit test and deploy application on server.

- CRA makes use of Enzyme and Jest for Unit testing. Does not need separate Jest installation or Web pack tweaking. Everything is taken care by CRA.

# Salient features (continued)

- Well thought React components to modularize code.
- Made use of Typescript and React Typescript support to develop generic component that’s used in all screen to automate task like showing Page loading Icon, display UI when data is available and show error if any.
- The above component has been reused in all screens of ReactJS PoC application.

# Salient features (continued)

- Role based authorization further limits access to screens and components use for creating RiskTypes.
- User can add RiskType and associated RiskTypeFields in one go using header and details table in screen.
- User can add Risk and Risk Fields in one go using header and details table in screen.

# Salient features (continued)

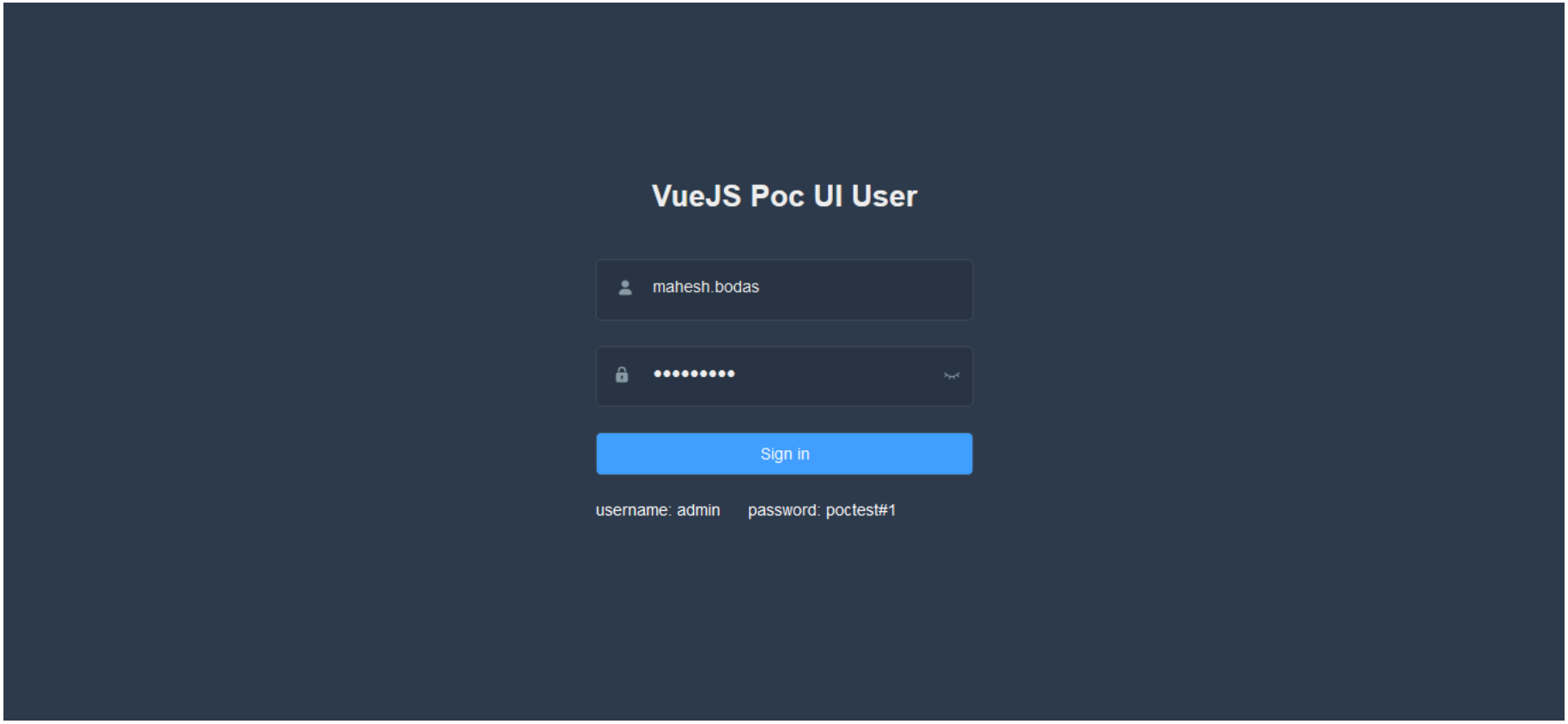
- Create Risk Instance screen collect data by dynamically adding controls related to various Risk type fields.
- Create Risk Instance screen dynamically adds required field validation and field type specific validation for controls related to Risk type fields. i.e. Date / float / integer.
- Successful model creation messages are displayed using MessageBox.
- Display various model validation errors using MessageBox

# Salient features (continued)

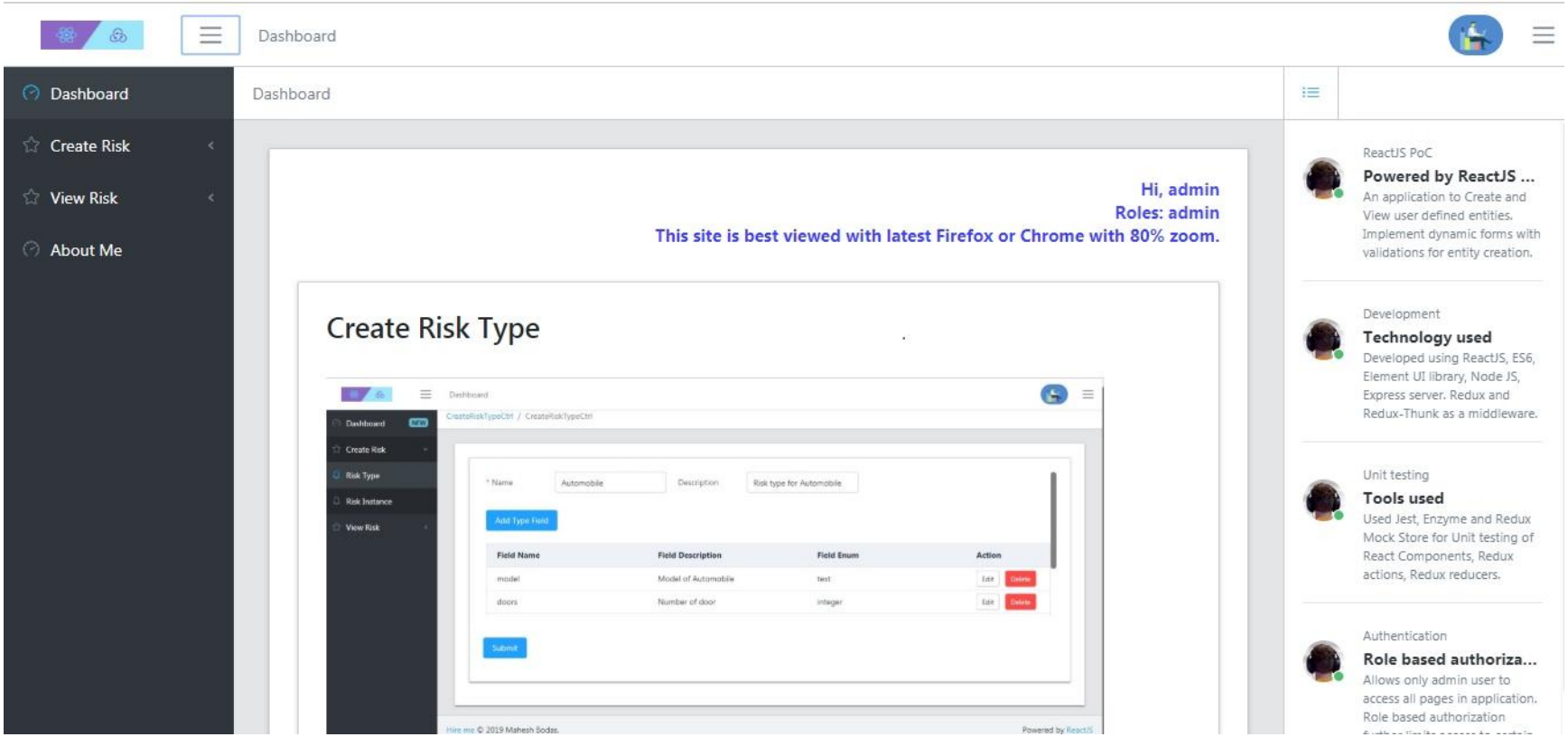
- View Single Risk allows user to fetch details for given Risk and renders various read-only controls related to Risk Type fields.

- View All Risks allow user to view All Risks entered in system based on RiskType filter.

# Login Screen



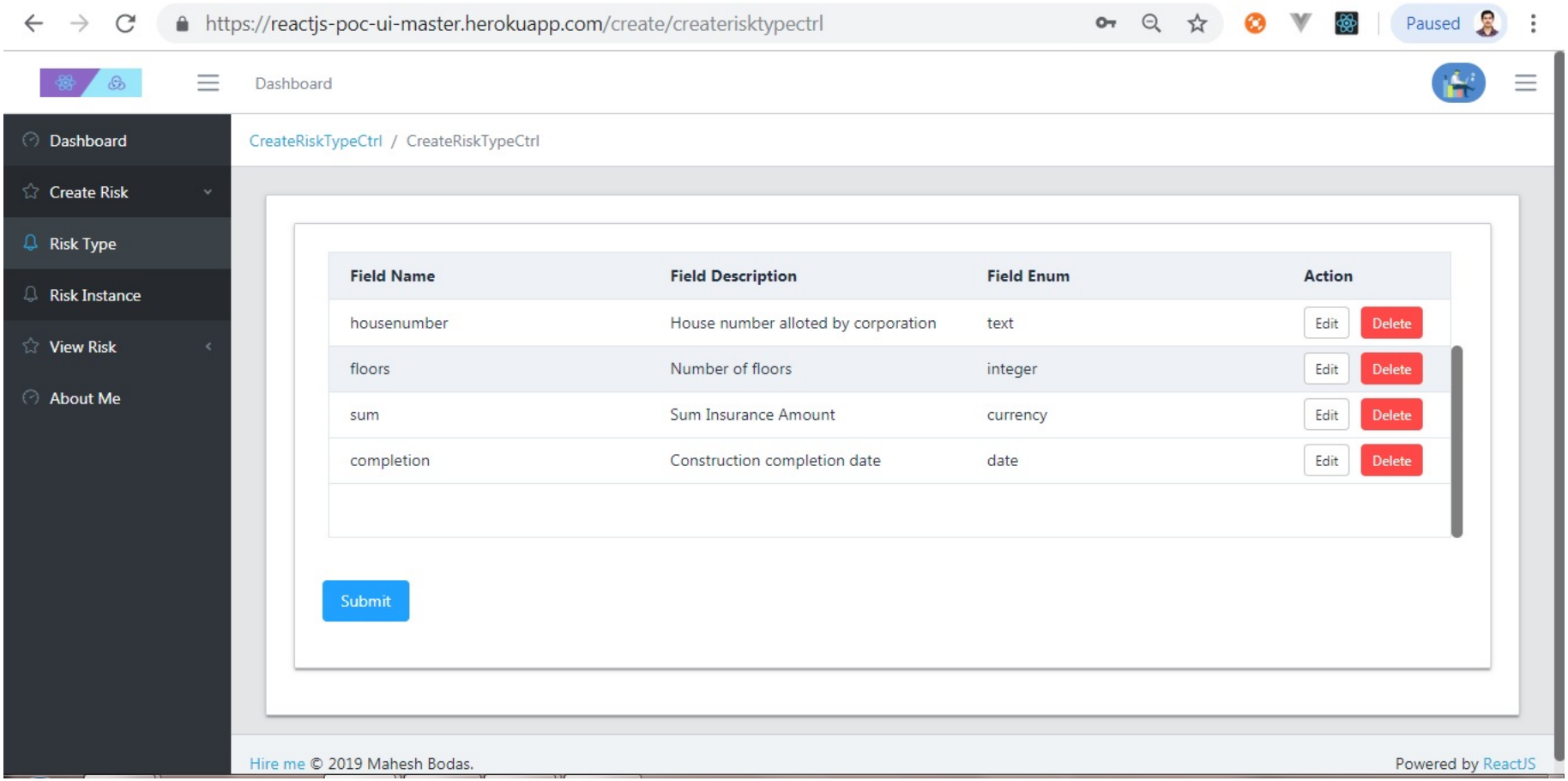
# Dashboard showing app highlights



# Create Risk Type screen

- Admin User can define Risk types.
- User Enters Risk type name and description.
- Click Add Risk Type Field button to define Risk type fields.
- Post Risk type after adding / editing Risk type fields.

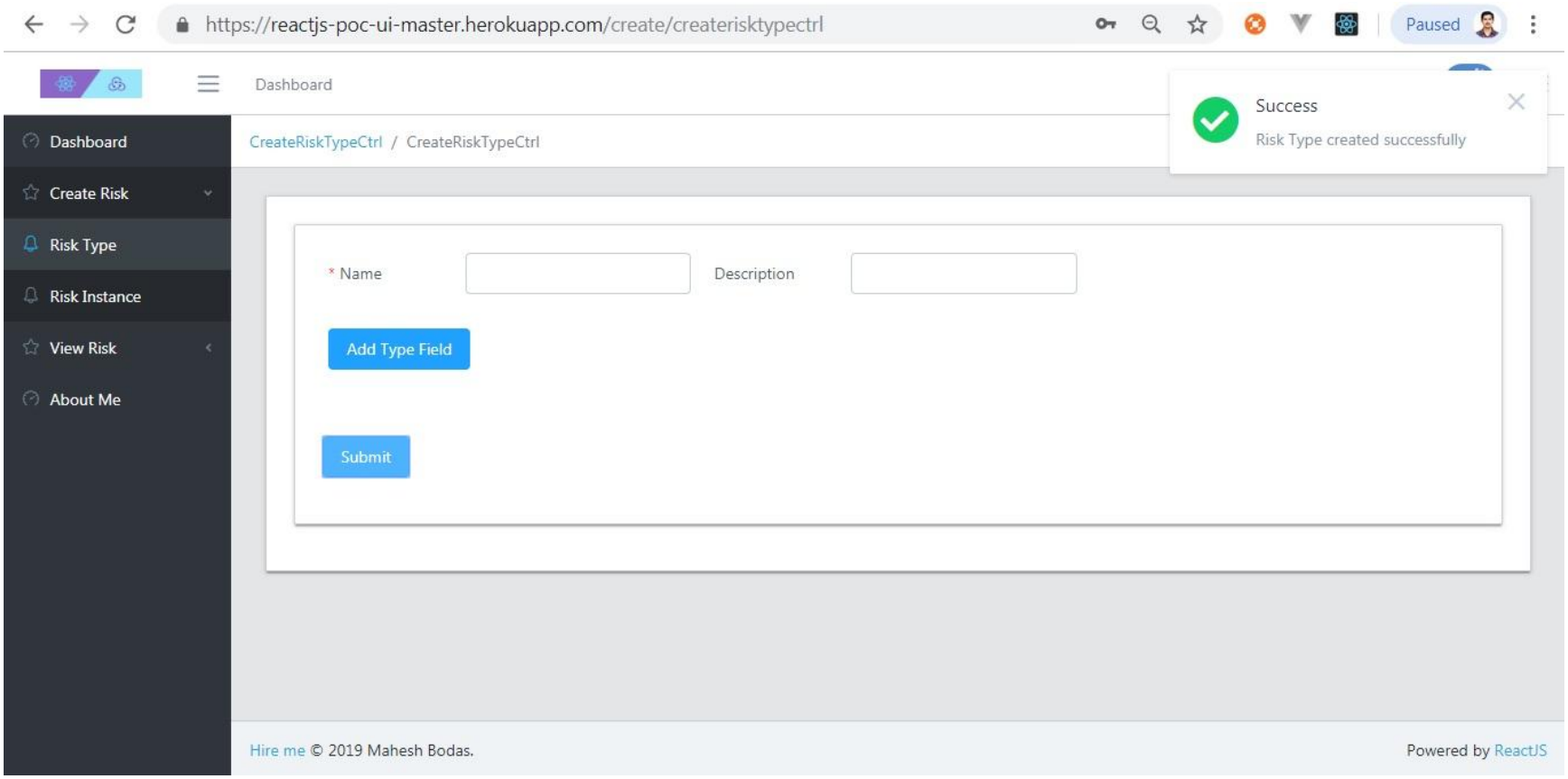
# Create Risk Type Image



# Risk Type Data posted to server


```
{"risk_type_name":"Home","risk_type_description":"Risk type for Home","risktype_risktypefields":[{
"risk_type_field_name":"housenumber","risk_type_field_enum":"text","risk_type_field_description":"House Number"
},{"risk_type_field_name":"floors","risk_type_field_enum":"integer","risk_type_field_description":"Number of
floors"},{"risk_type_field_name":"sum","risk_type_field_enum":"currency","risk_type_field_description":
"Insurance amount"},{"risk_type_field_name":"completion","risk_type_field_enum":"date",
"risk type field description":"Date of completion"}]}
```

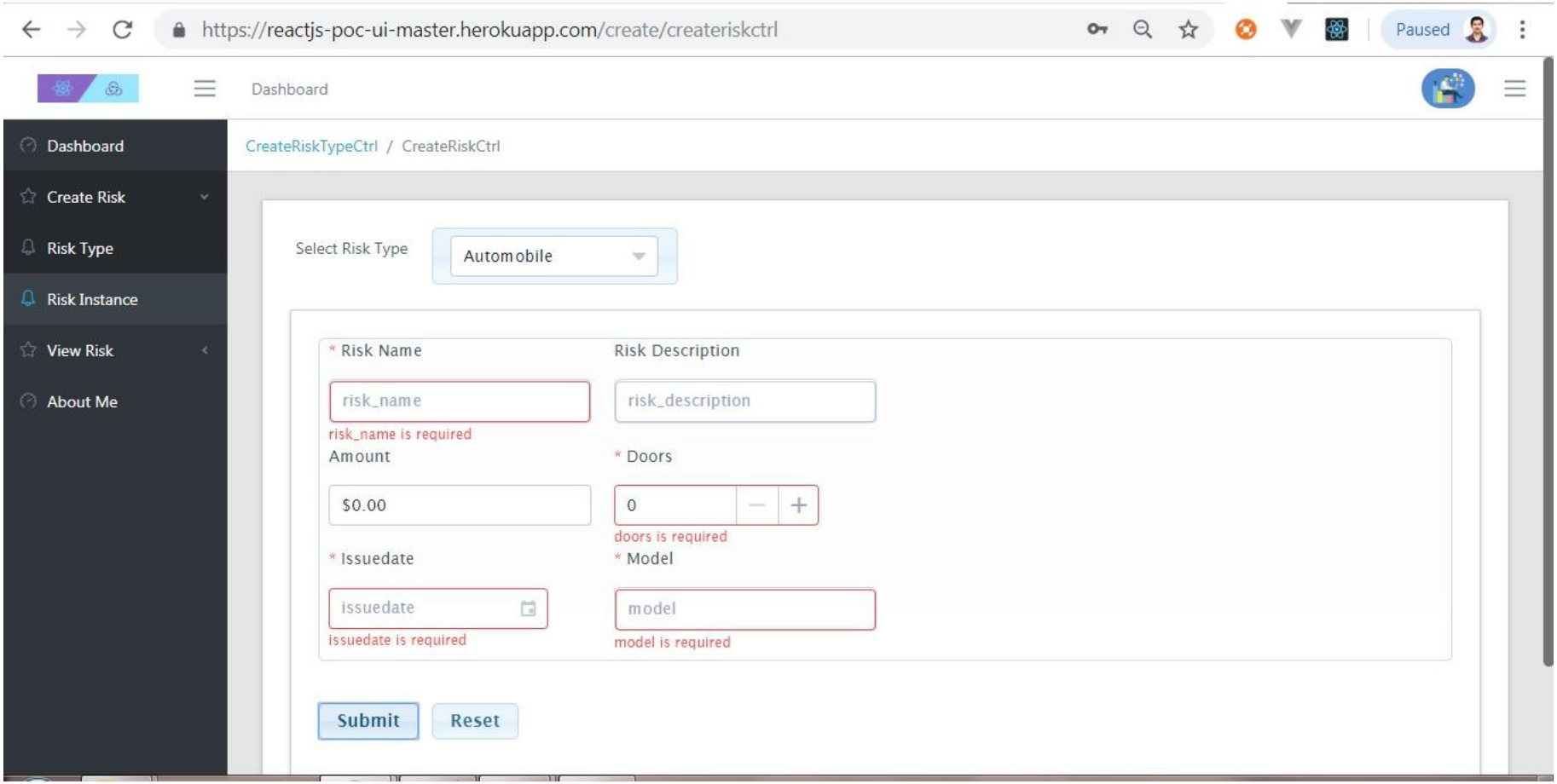
# Create RiskType shows Success



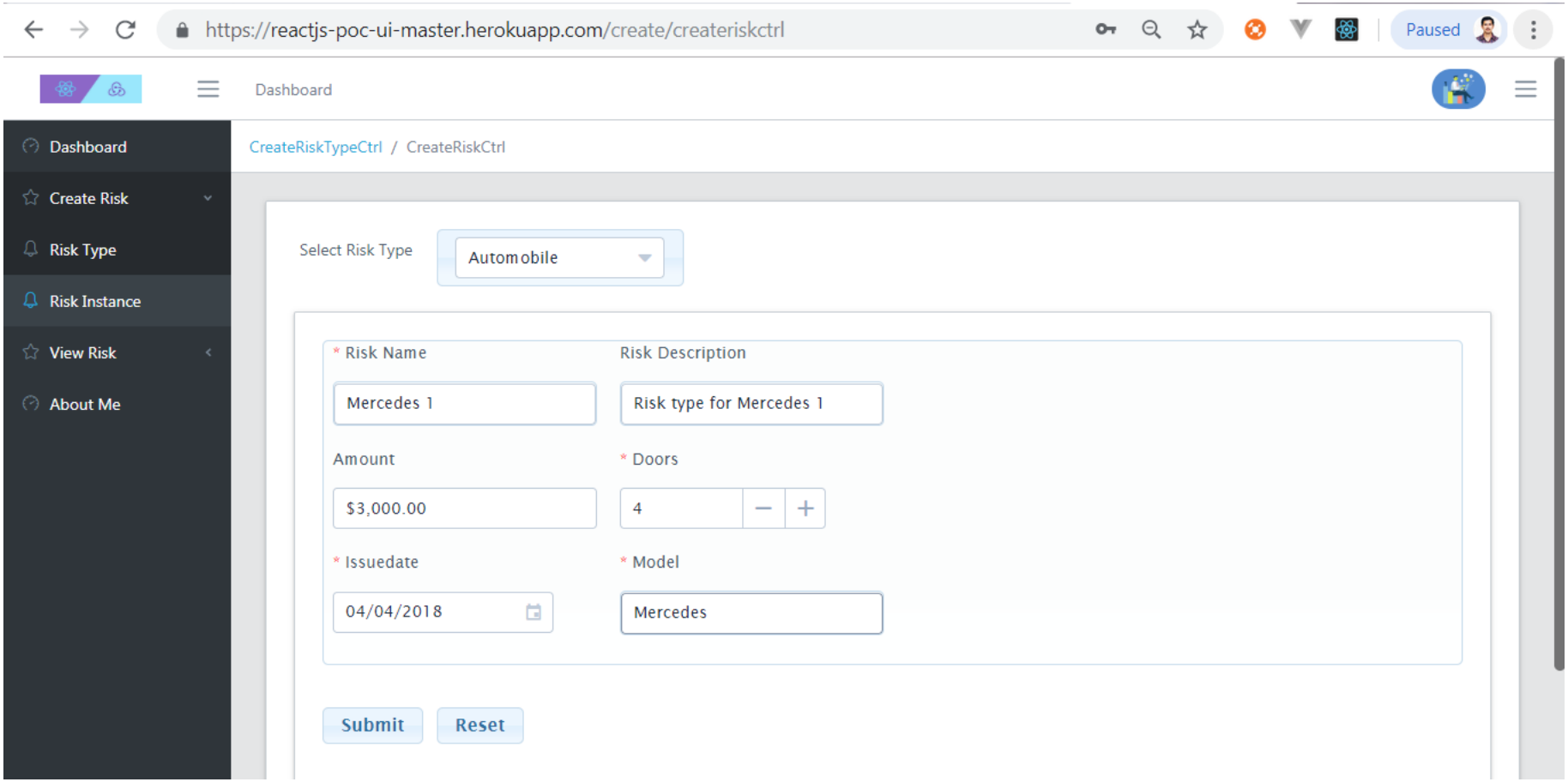
# Create Risk Instance screen.

- User can create Risk Instance based on Risk types.
- Select appropriate Risk Type from dropdown box.
- Fill up all Risk Instance fields and submit form.
- Admin user can define Risk Types.

# Screen showing field validation errors



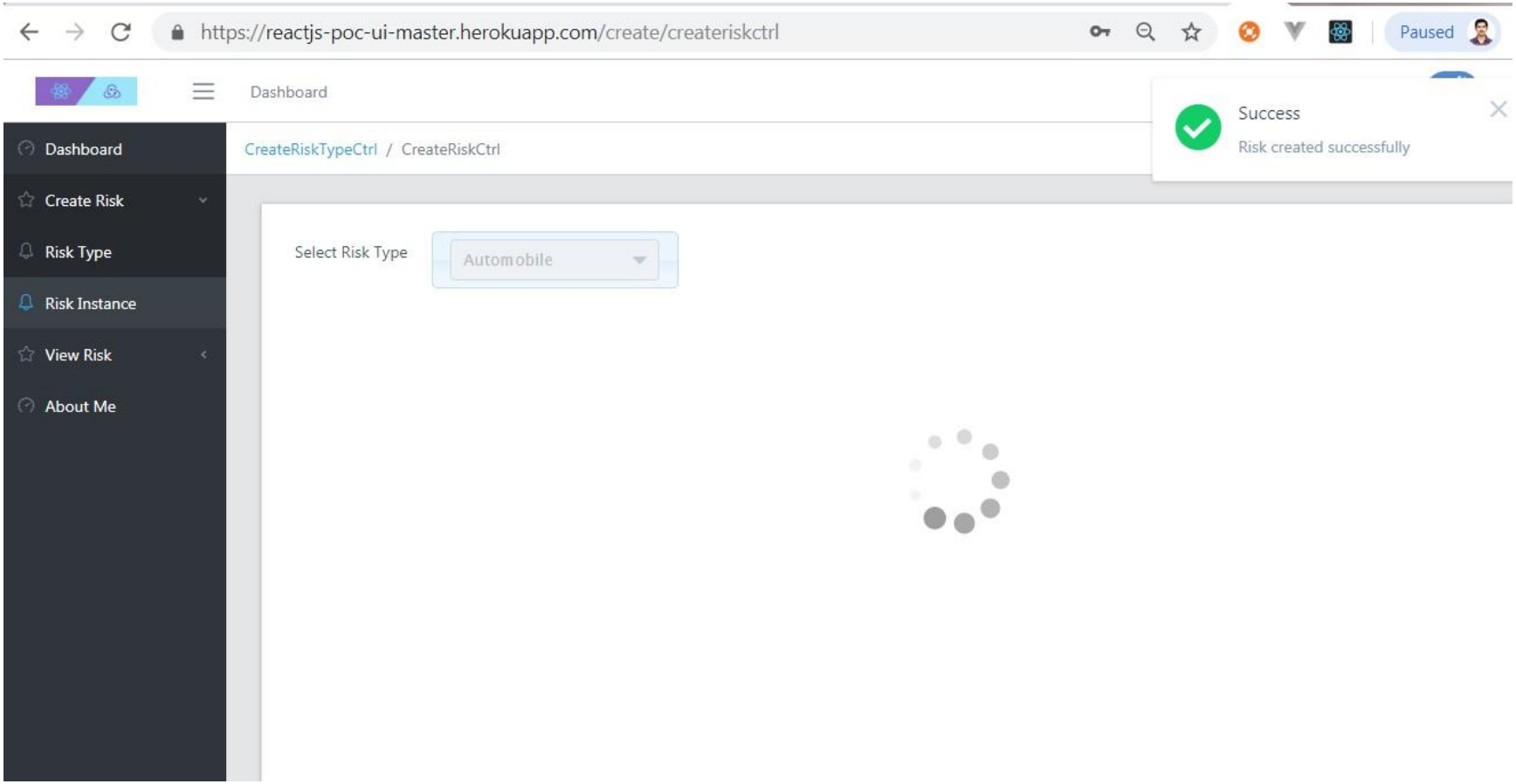
# Submit when validation succeed



# Posting Risk Instance data to server


```
{"risktype":1,"risk_name":"Mercedes 1","risk_description":"Insurance for Mercedes","risk_riskfields":[{
"risktypefield":4,"risk_field_value":"07/14/2016"},{"risktypefield":3,"risk_field_value":30000},{"risktypefield":
2,"risk_field_value":4},{"risktypefield":1,"risk_field_value":"Mercedes"}]}
```

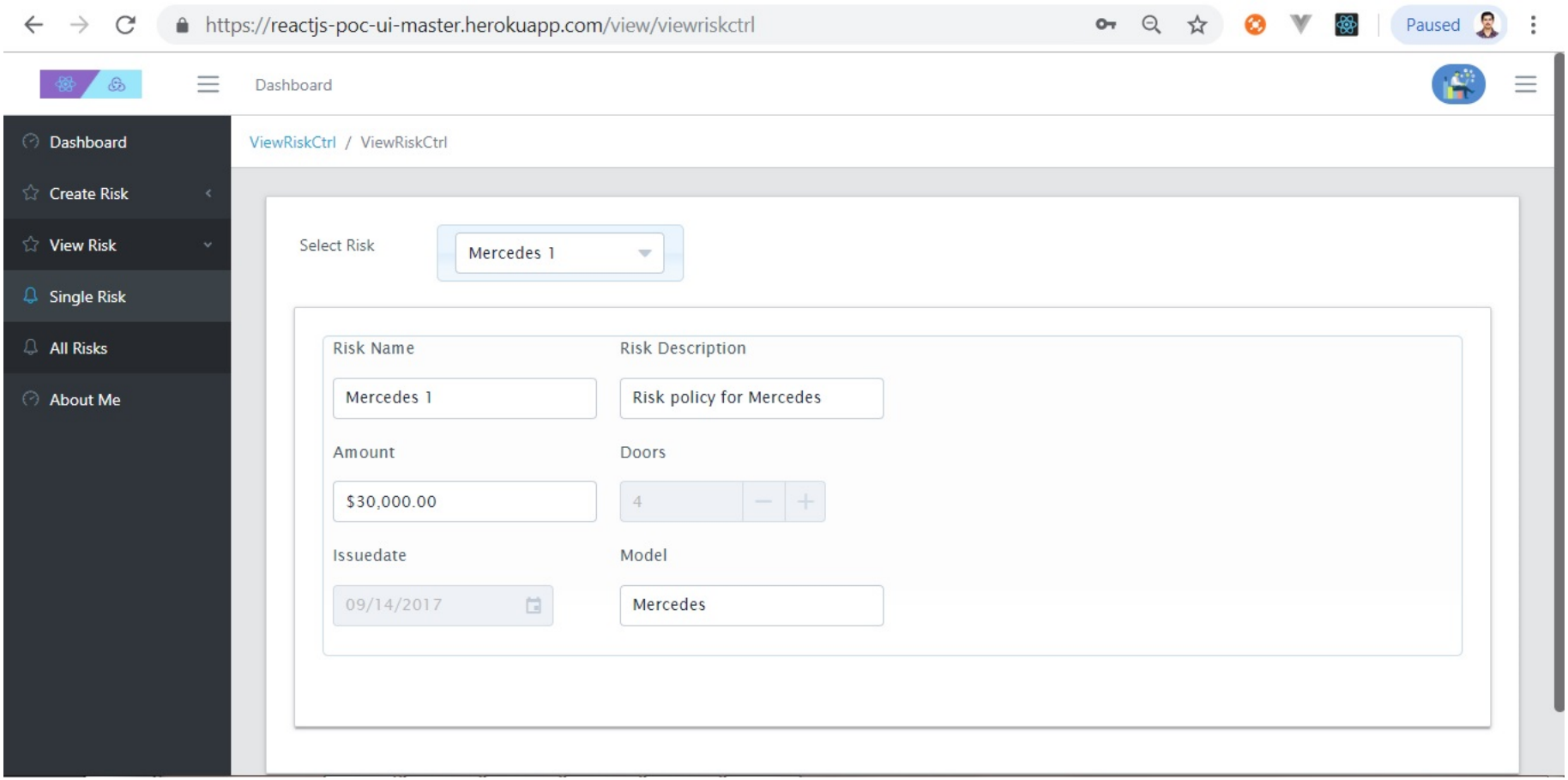
# Create Risk Instance showing success



# View Single Risk Instance

- User can view single Risk Instance using this form.
- Select appropriate Risk Instance from dropdown box.
- Screen will show Risk name and all Risk Instance fields (read-only).

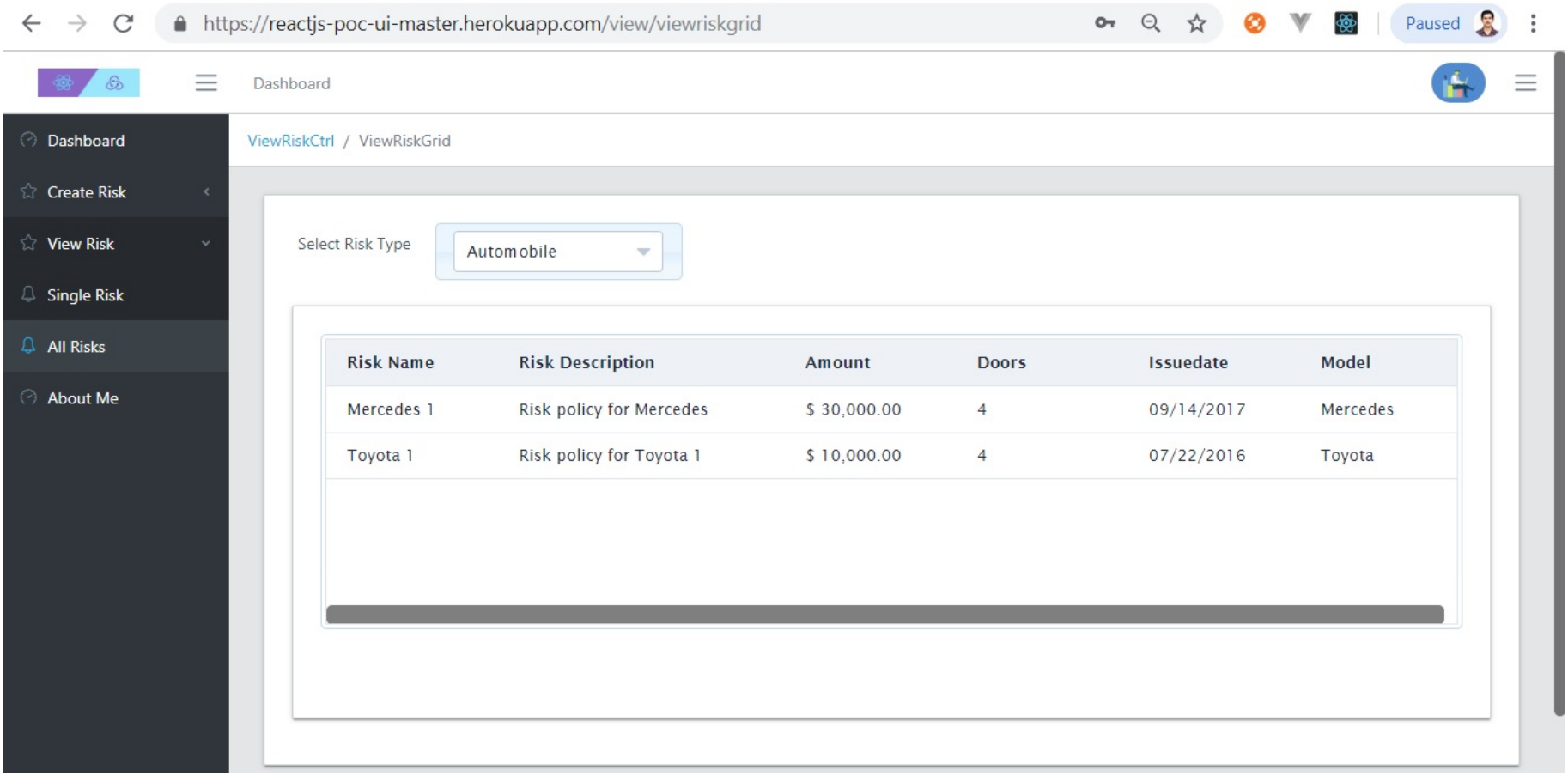
# View Single Risk Screen



# View All Risks

- User can view Risk Instances using this form
- Select appropriate Risk Type from dropdown box.
- Screen will show all Risks and associated instance fields in tabular format.

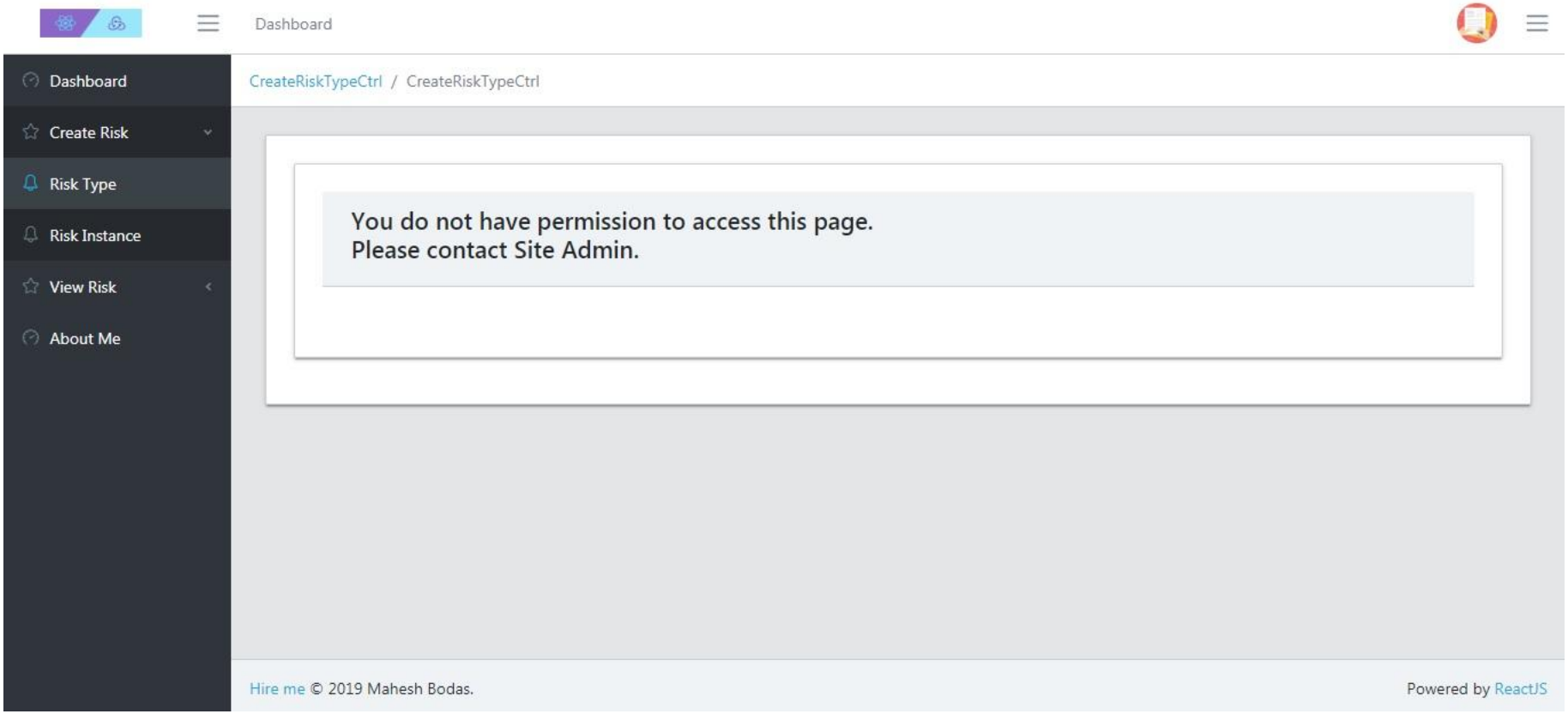
# View All Risk Screen



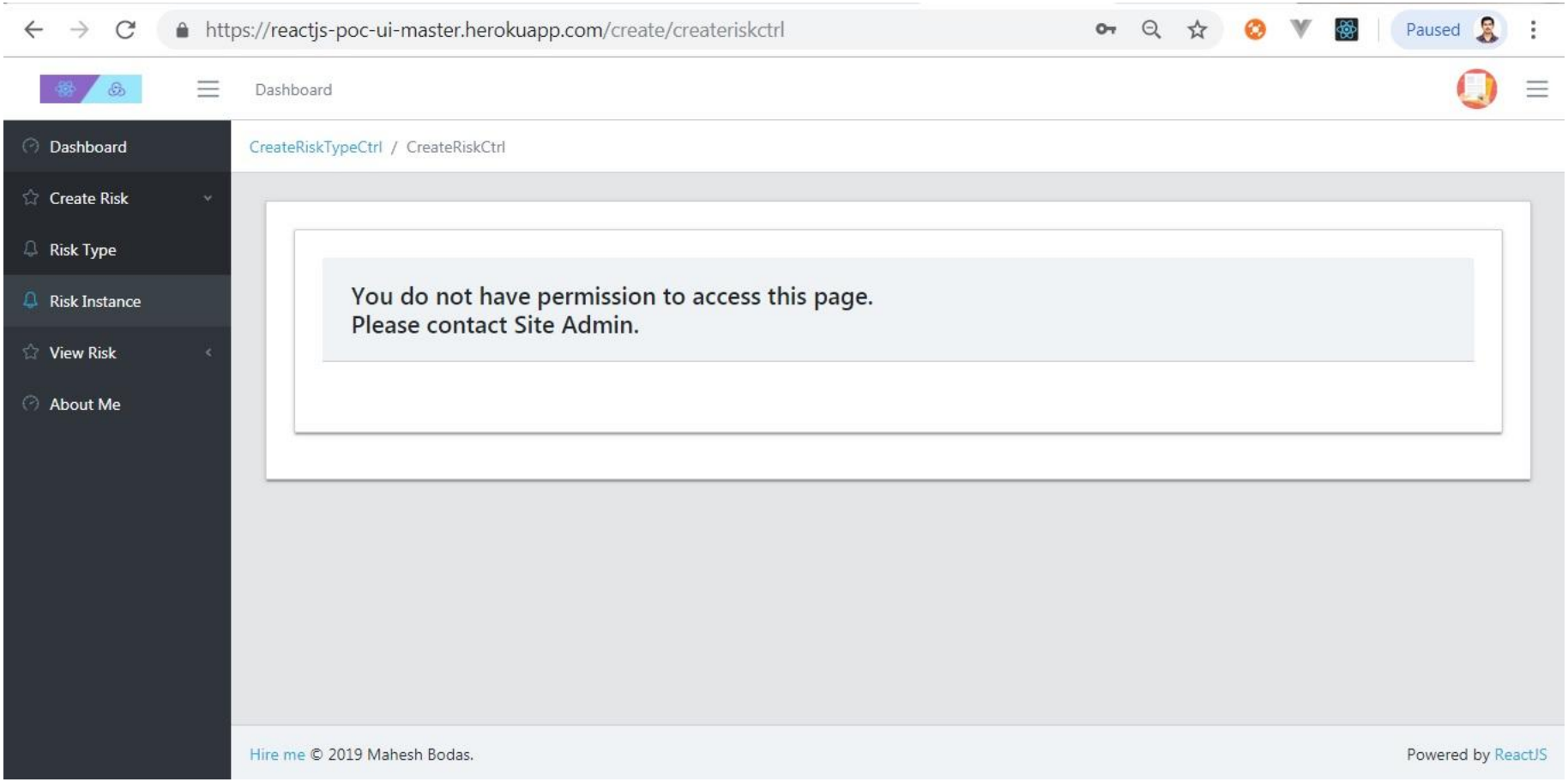
# Non Admin users

- All above slides assumed user logged in has admin privilege.
- Non admin user will be restricted from accessing Create Risk Type and Create Risk Instance page.
- They can view Risk Instances added to system.
- User : editor (lower case)
- Password : test#123 (lower case)

# No access to Create Risk Type



# No access to Create Risk Instance



# Unit testing.

- Used Enzyme, Jest for unit testing React components.
- Used Redux Mock Store for Unit testing of, Redux actions, Redux reducers.

# Known issues.

- Since Django Rest API and UI application are installed using free account on Heroku platform.
- In case if  Django Rest API is not invoked in last one hour or more Dyno assciated with it goes to sleep.
- As a result it may be possible that when UI invokes API it might timeout.
- In case if you get any error try refreshing page after 2/3 minutes.

# Known issue (continued)